

```
In [2]: import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
```

```
In [5]: # Loading the csv data to a Pandas DataFrame
heart_data = pd.read_excel('F:\Final year project\INTERNSHIP\heart disease.xlsx')
# Read Raw Dataset
heart_data.head()
```

```
Out[5]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
In [6]: # print Last 5 rows of the dataset
heart_data.tail()
```

```
Out[6]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

```
In [7]: # number of rows and columns in the dataset
heart_data.shape
```

```
Out[7]: (303, 14)
```

```
In [8]: # getting some info about the data
heart_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   age         303 non-null    int64
1   sex         303 non-null    int64
2   cp          303 non-null    int64
3   trestbps    303 non-null    int64
4   chol        303 non-null    int64
5   fbs         303 non-null    int64
6   restecg     303 non-null    int64
7   thalach     303 non-null    int64
8   exang       303 non-null    int64
9   oldpeak     303 non-null    float64
10  slope       303 non-null    int64
11  ca          303 non-null    int64
12  thal        303 non-null    int64
13  target      303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

```
In [9]: # checking for missing values
heart_data.isnull().sum()
```

```
Out[9]: age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

```
In [10]: # statistical measures about the data
heart_data.describe()
```

Out[10]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.591254
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.486590
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000

```
In [11]: # checking the distribution of Target Variable
heart_data['target'].value_counts()
```

```
Out[11]: 1    165
         0    138
         Name: target, dtype: int64
```

```
In [12]: X = heart_data.drop(columns='target', axis=1)
         Y = heart_data['target']
         print(X)
```

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0    63   1   3    145    233   1         0    150     0     2.3
1    37   1   2    130    250   0         1    187     0     3.5
2    41   0   1    130    204   0         0    172     0     1.4
3    56   1   1    120    236   0         1    178     0     0.8
4    57   0   0    120    354   0         1    163     1     0.6
..    ...  ..  ..    ...    ...  ...    ...    ...    ...    ...
298   57   0   0    140    241   0         1    123     1     0.2
299   45   1   3    110    264   0         1    132     0     1.2
300   68   1   0    144    193   1         1    141     0     3.4
301   57   1   0    130    131   0         1    115     1     1.2
302   57   0   1    130    236   0         0    174     0     0.0

   slope  ca  thal
0       0   0    1
1       0   0    2
2       2   0    2
3       2   0    2
4       2   0    2
..    ...  ..  ...
298     1   0    3
299     1   0    3
300     1   2    3
301     1   1    3
302     1   1    2
```

[303 rows x 13 columns]

```
In [13]: print(Y)
```

```
0     1
1     1
2     1
3     1
4     1
..
298   0
299   0
300   0
301   0
302   0
Name: target, Length: 303, dtype: int64
```

```
In [19]: heart_data['target'].value_counts()
```

```
Out[19]: 1    165
         0    138
         Name: target, dtype: int64
```

```
In [20]: heart_data.groupby('target').mean()
```

```
Out[20]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
target													
0	56.601449	0.826087	0.478261	134.398551	251.086957	0.159420	0.449275	139.101449	0.550725	1.0			
1	52.496970	0.563636	1.375758	129.303030	242.230303	0.139394	0.593939	158.466667	0.139394	0.0			

```
In [21]: X = heart_data.drop(columns = 'target', axis=1)
Y = heart_data['target']
```

```
In [22]: print(X)
```

```

   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0    63   1   3     145    233   1         0     150      0      2.3
1    37   1   2     130    250   0         1     187      0      3.5
2    41   0   1     130    204   0         0     172      0      1.4
3    56   1   1     120    236   0         1     178      0      0.8
4    57   0   0     120    354   0         1     163      1      0.6
..   ...  ..  ..     ...    ...   ...      ...     ...     ...     ...
298  57   0   0     140    241   0         1     123      1      0.2
299  45   1   3     110    264   0         1     132      0      1.2
300  68   1   0     144    193   1         1     141      0      3.4
301  57   1   0     130    131   0         1     115      1      1.2
302  57   0   1     130    236   0         0     174      0      0.0

   slope  ca  thal
0      0   0    1
1      0   0    2
2      2   0    2
3      2   0    2
4      2   0    2
..   ...  ..  ...
298    1   0    3
299    1   0    3
300    1   2    3
301    1   1    3
302    1   1    2

```

[303 rows x 13 columns]

```
In [23]: print(Y)
```

```

0      1
1      1
2      1
3      1
4      1
..
298    0
299    0
300    0
301    0
302    0
Name: target, Length: 303, dtype: int64

```

```
In [24]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=1)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(303, 13) (242, 13) (61, 13)
```

```
In [25]: classifier = svm.SVC(kernel='linear')
```

```
In [26]: #training the support vector Machine Classifier  
classifier.fit(X_train, Y_train)
```

```
Out[26]: SVC(kernel='linear')
```

```
In [27]: # accuracy score on the training data  
X_train_prediction = classifier.predict(X_train)  
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
In [28]: print('Accuracy score of the training data : ', training_data_accuracy)
```

```
Accuracy score of the training data : 0.8553719008264463
```

```
In [29]: # accuracy score on the test data  
X_test_prediction = classifier.predict(X_test)  
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)  
print('Accuracy score of the test data : ', test_data_accuracy)
```

```
Accuracy score of the test data : 0.819672131147541
```

```
In [33]: input_data = (56,1,1,120,236,0,1,178,0,0.8,2,0,2)  
  
    # changing the input_data to numpy array  
    input_data_as_numpy_array = np.asarray(input_data)  
  
    # reshape the array as we are predicting for one instance  
    input_data_resaped = input_data_as_numpy_array.reshape(1,-1)  
  
    prediction = classifier.predict(input_data_resaped)  
    print(prediction)  
  
    if (prediction[0] == 0):  
        print('The person doesnt have heart disease')  
    else:  
        print('The person has heart disease')
```

```
[1]  
The person has heart disease
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not  
have valid feature names, but SVC was fitted with feature names  
... warnings.warn(
```